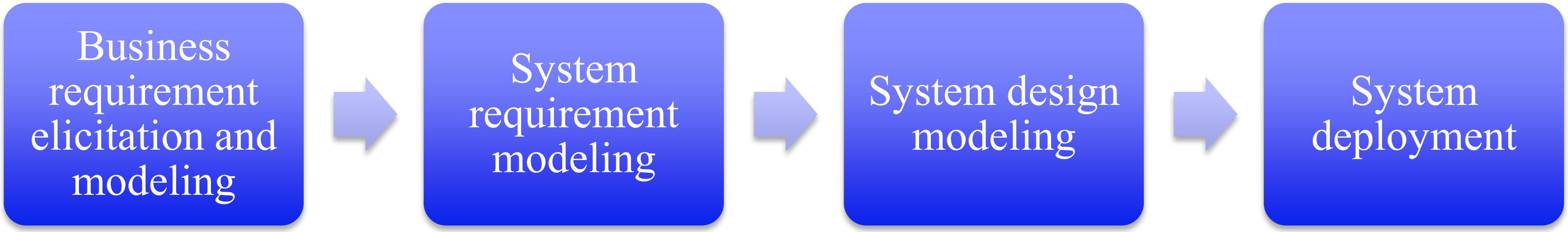
UOW

User Stories and Use Cases

CSIT114 / 814: Systems Analysis

UNIVERSITY
OF WOLLONGONG
AUSTRALIA



Business requirement elicitation and modeling → System requirement modeling → System design modeling → System deployment

- **Who?**
  Stakeholder
  System analysts
- **What?**
  SDLC
  PM
- **How?**
  Method

- **How to gather requirement?**
  - Survey, Interview
- **Activity, use case, sequence and domain diagrams**

- **Detailed design model**
- **Architecture design model**

- **Principles of system deployment**

Overview

- User Stories
- Use Cases
- The User Goal Technique and Personas
- The Event Decomposition Technique
- Use Case Description and Activity Diagram

User Stories

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

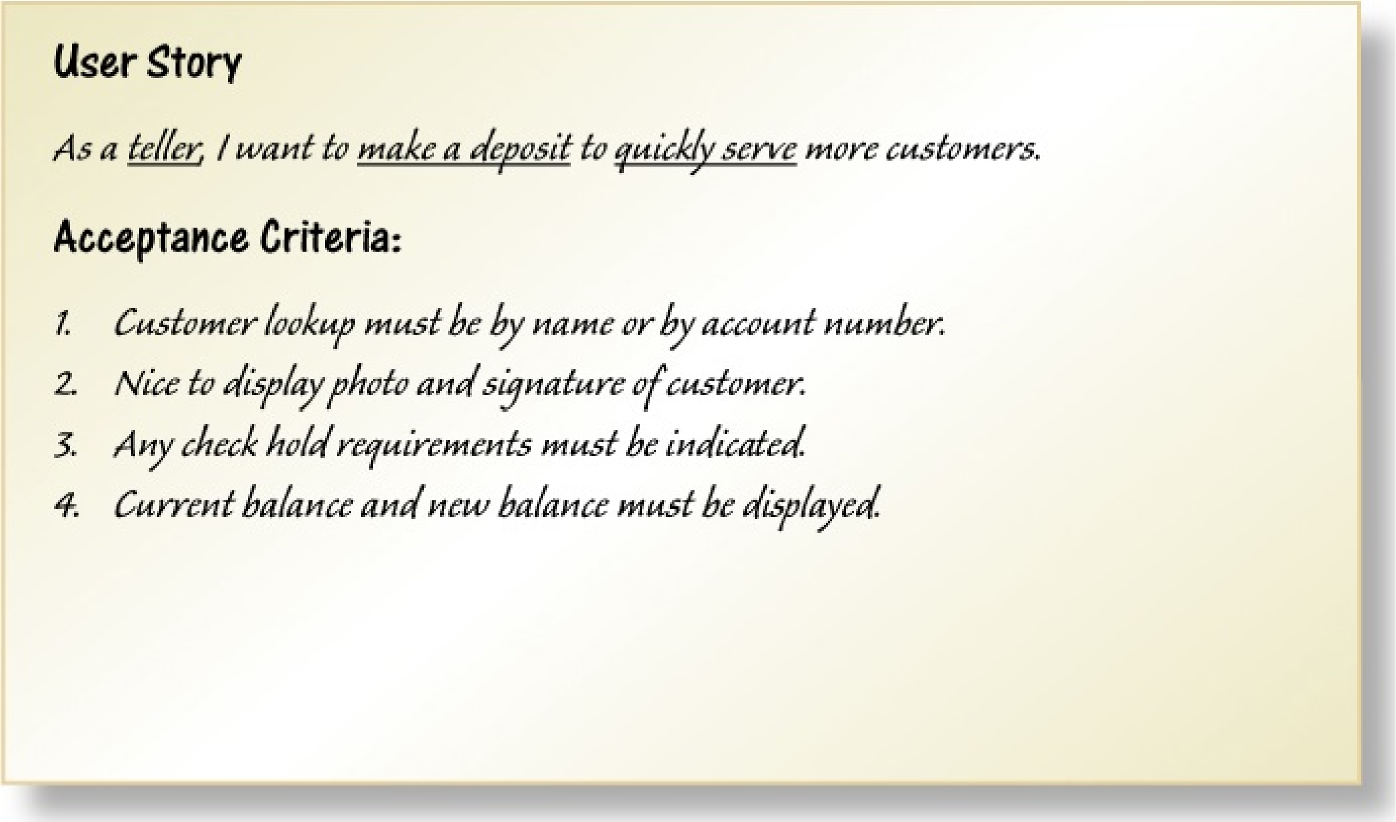
User Stories

- A User Story is a one-sentence description of a *work-related task done by a user to achieve some goal or result*
- Suitable in Agile development
  - Simplicity, value added and user collaboration
  - Less formal than UML diagrams
- The template for a user story description is:
  - “As a <role> I want to <goal> so that <benefit>”

User Stories

- **Acceptance Criteria** identify the features that must be present at the completion of the task
  - For testing (i.e. Beta testing) once user stories are implemented
  - Focused on functionality (functional requirements)
  - For verifying whether users are looking at the user stories they provide at an appropriate level

Sample User Story



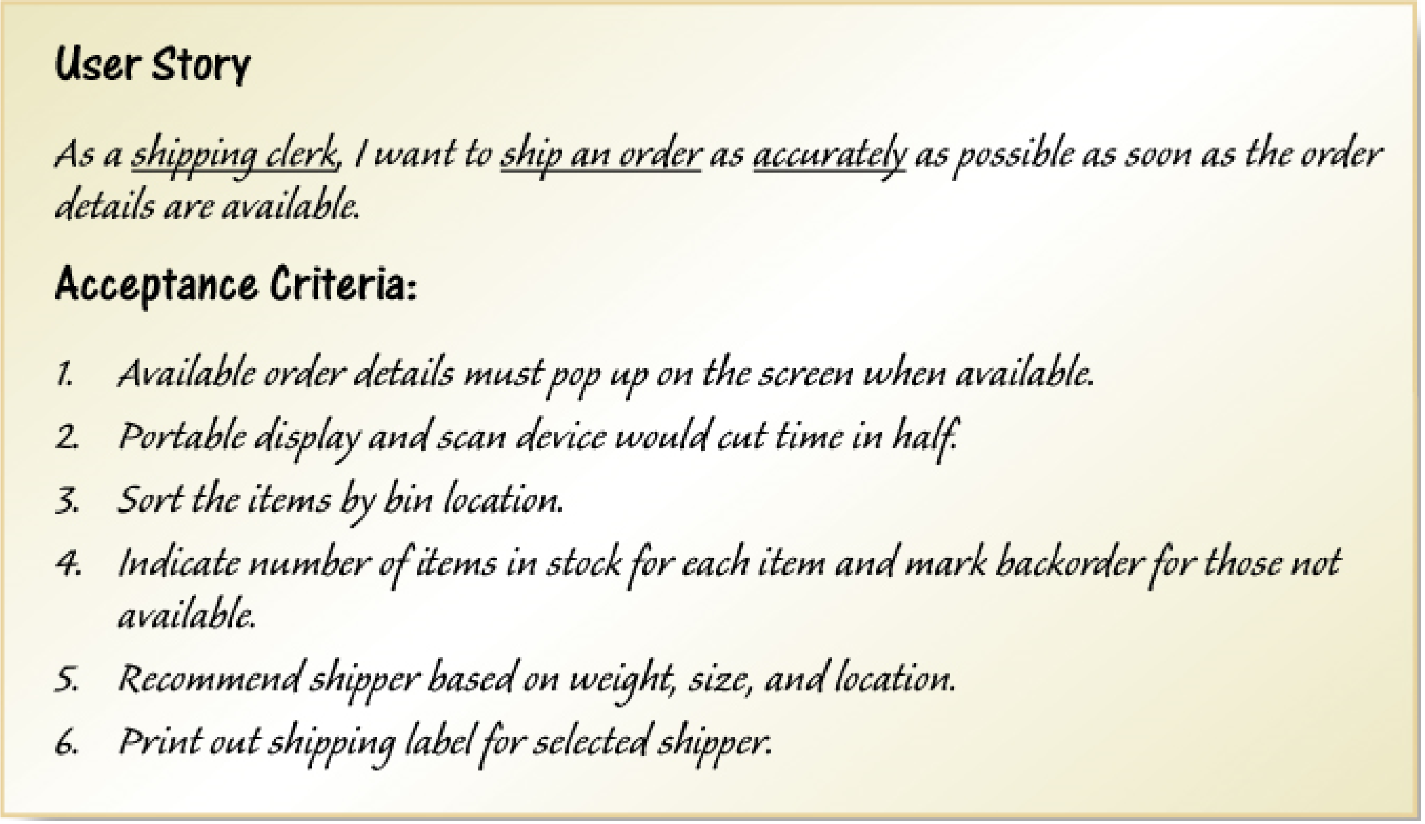
User Story

As a teller, I want to make a deposit to quickly serve more customers.

Acceptance Criteria:

1. Customer lookup must be by name or by account number.
2. Nice to display photo and signature of customer.
3. Any check hold requirements must be indicated.
4. Current balance and new balance must be displayed.

Sample User Story



User Story

As a shipping clerk, I want to ship an order as accurately as possible as soon as the order details are available.

Acceptance Criteria:

1. *Available order details must pop up on the screen when available.*
2. *Portable display and scan device would cut time in half.*
3. *Sort the items by bin location.*
4. *Indicate number of items in stock for each item and mark backorder for those not available.*
5. *Recommend shipper based on weight, size, and location.*
6. *Print out shipping label for selected shipper.*

Use Cases

UNIVERSITY
OF WOLLONGONG
AUSTRALIA

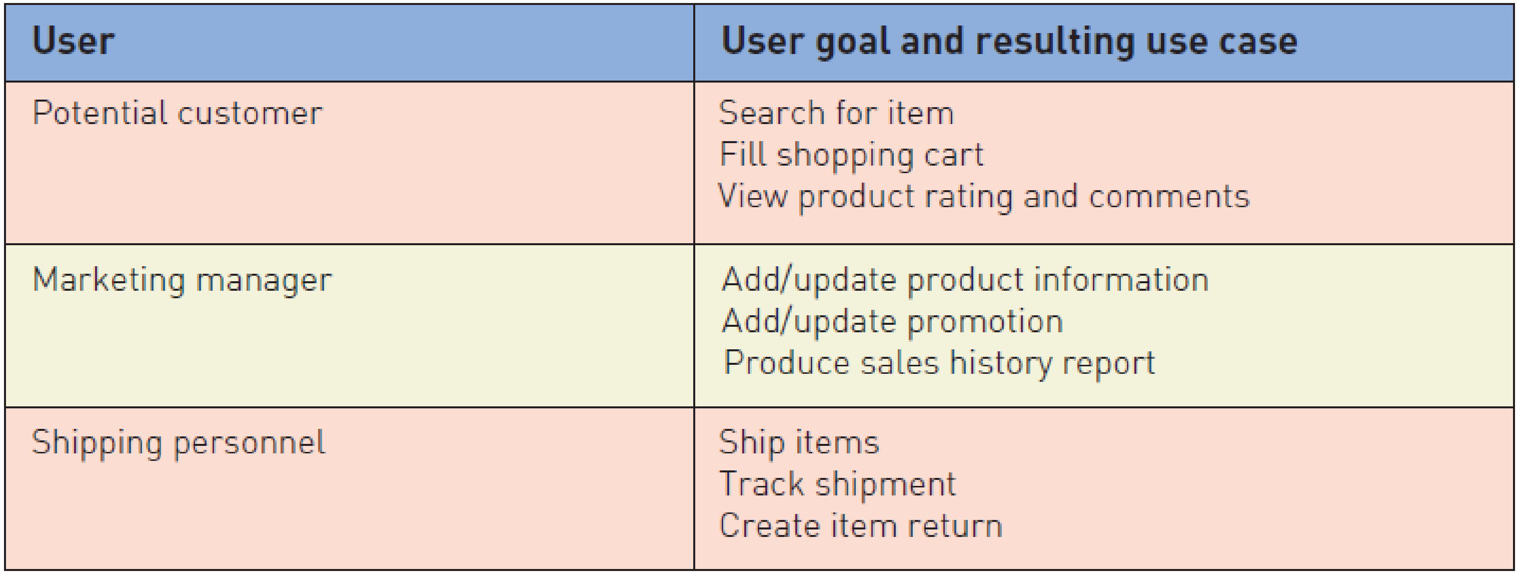
Use Cases

- **Use case** — *an activity that the system performs, usually in response to a request by a user*
  - Use cases define functional requirements
  - Analysts decompose the system into a set of use cases (functional decomposition, aka requirement traceability)
  - More formal than user stories
- Name each use case using Verb-Noun
  - E.g. look up supplier
- Two techniques for Identifying use cases
  - **User goal technique**
  - **Event decomposition technique**

User Goal Technique

- A technique to identify use cases by determining what specific goals or objectives must be completed by the system for the user.
- This technique is the most common in industry
- Simple and effective
- Identify all of the potential categories of users of the system
- Interview and ask them to describe the tasks the computer can help them with
- Probe further to refine the tasks into specific user goals

User Goal Technique - Examples



User	User goal and resulting use case
Potential customer	Search for item Fill shopping cart View product rating and comments
Marketing manager	Add/update product information Add/update promotion Produce sales history report
Shipping personnel	Ship items Track shipment Create item return

User Goal Technique: Specific Steps

1. Identify all the potential users for the new system
2. Classify the potential users in terms of their *functional role* (e.g., shipping, marketing, sales)
3. Further classify potential users by *organisational level* (e.g., operational, management, executive)
4. For each type of user, interview them to find a list of specific goals they will have when using the new system
   - current goals and innovative functions to add value

User Goal Technique: Specific Steps

5. Create a list of preliminary use cases organised by type of user
6. Look for duplicates with similar use case names and resolve inconsistencies
7. Identify where different types of users need the same use cases
8. Review the completed list with each type of user and then with interested stakeholders

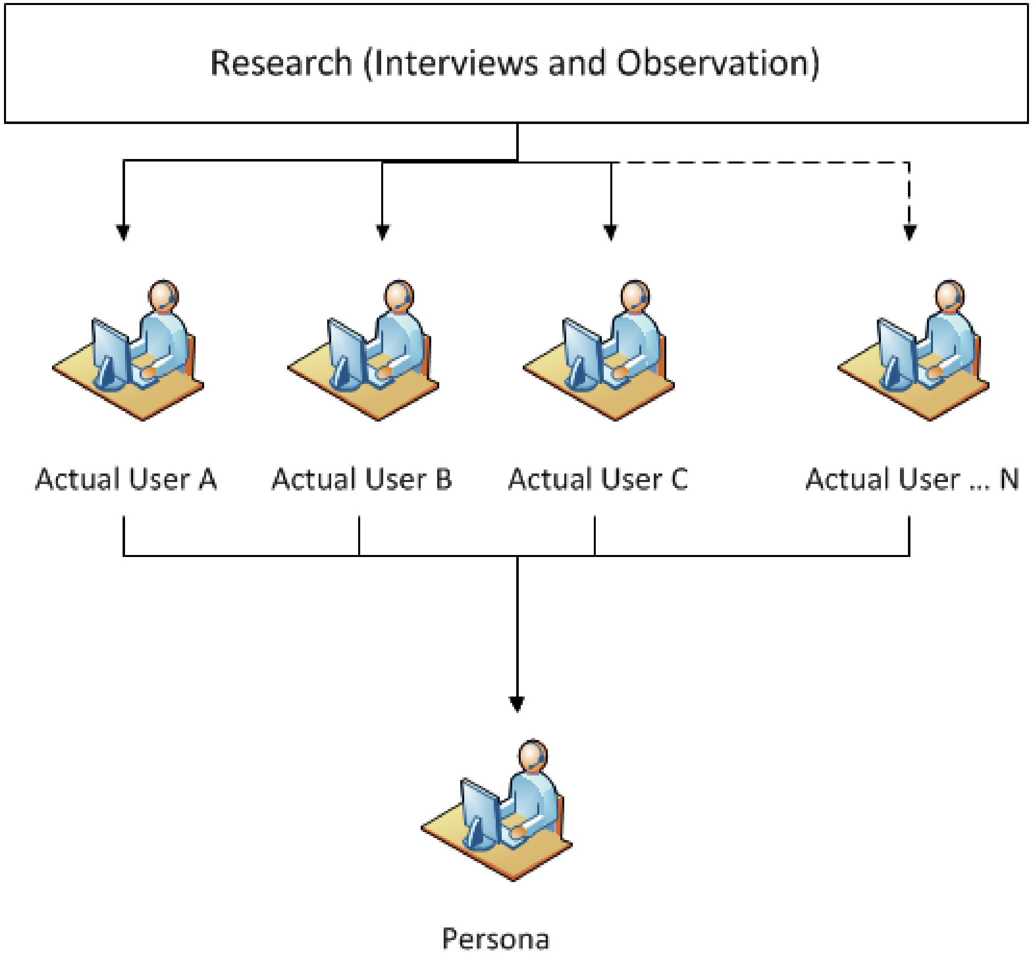
User Goal Technique

- User type identification
  - Functional role
  - Organisational role
- *What if multiple users in one role?*
  - Each of them may have different usage scenarios
  - *The **Persona** theory*

About Personas

- Personas are fictional archetypes based on research into real users
  - Combination of understanding multiple users
- Archetype not a stereotype
  - Archetype – a typical example of a person or object
  - Stereotype – an oversimplified idea of a person or object
- Advantages
  - Focus on one "canonical" example rather than complimentary and possibly inconsistent individuals
  - Improve the understanding of "the" use case
  - Facilitate communication and commitment of system functionalities

Persona Development



Persona Risks

Incorrect persona development due to

- Insufficient buy-in of personas by the project team and stakeholders
- Not enough user-provided information or feedback (e.g., in the form of user stories)
- Poor understanding of users

Goals and Personas (Cooper, 2007)

A theory for sufficient understanding on users:

- Understanding user goals are essential to develop products
- To be effective personas must have goals based on the real goals (from interviews)
- Alan Cooper identifies 3 types of user goals and 5 types of personas

Types of user goals (assumptions)

Type of goal	Examples	Use in persona
Life goal - The goals a user has for their life	▪ To succeed ▪ To enjoy life ▪ Behave ethically	▪ Little direct use ▪ However, if a system can help achieve these goals it will be more interesting to the user
Experience goals - How the user wants their experiences to be in general (and so with the system). *This is their definition of a good experience*	▪ Not feel stupid ▪ Not make mistakes ▪ Enjoy using the product	▪ These are very important as these define the user experience with the system
End goals - The goals that user want to achieve by using the system	▪ To stay in contact with colleagues ▪ To know what your next appointment is	▪ Very important ▪ If your system does not achieve these, then it has failed

Persona types

Type	Description
Primary	The main target user for the product
Secondary	Other users who are mostly satisfied, but have some additional requirements
Supplemental	Not primary or secondary users, happy with the main product
Customer	The product customer (the buyer)
Served	Personas that do not use the product, but are served by the use of the product
Negative	Personas of the user *who will not be considered in the product design*

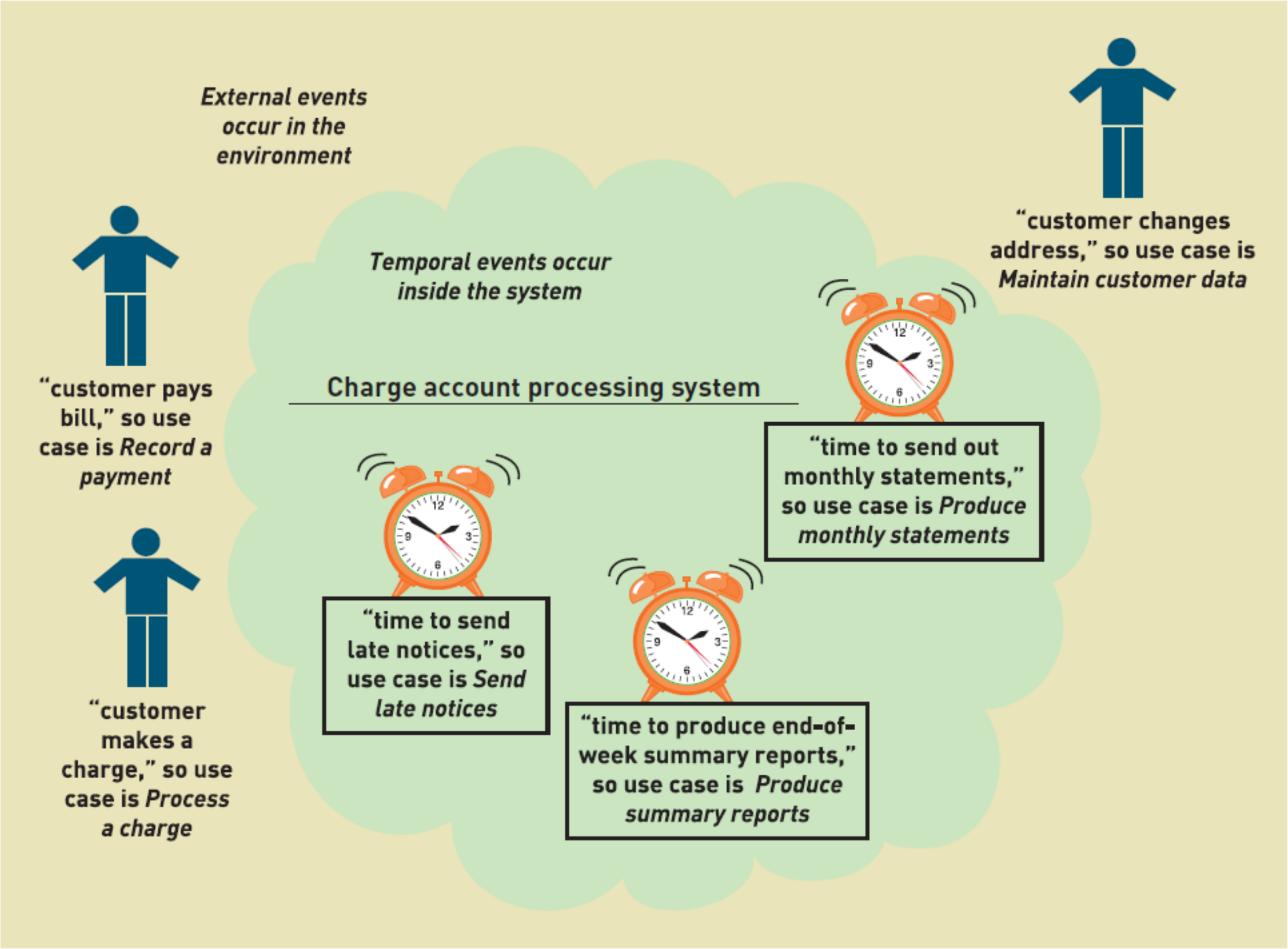
Event Decomposition Technique

- More Comprehensive and Complete Technique
  - Identify the events that occur to which the system must respond.
  - For each event, name a use case (verb-noun) that describes what the system does when the event occurs (but no need to go to the underlying functions).
- The appropriate level of details; for example,
  - Typing a customer name on the form as a use case?
  - A use case includes all working activities with customers, e.g., adding new customer accounts, updating records and contacting late-paying customers?
  - A use case is the entire process of adding a new customer account?

Event Decomposition Technique

- **Elementary business processes (EBP)**
  - A most fundamental task in a business process that is performed by one person in one place in response to a business event, adds measurable value, and leaves the system and its data in a stable and consistent state.
- **Event** – something that occurs at a specific time and place, can be described, and should be remembered by the system

Events and Use Cases



External events occur in the environment

"customer pays bill," so use case is *Record a payment*

"customer makes a charge," so use case is *Process a charge*

"customer changes address," so use case is *Maintain customer data*

Temporal events occur inside the system

Charge account processing system

"time to send late notices," so use case is *Send late notices*

"time to produce end-of-week summary reports," so use case is *Produce summary reports*

"time to send out monthly statements," so use case is *Produce monthly statements*

Types of Events

- External Event
  - an event that occurs outside the system, usually initiated by an external agent or actor
- Temporal Event
  - an event that occurs as a result of reaching a point in time
- State Event
  - an event that occurs when something happens inside the system that triggers some process
  - E.g., reorder point is reached for inventory item

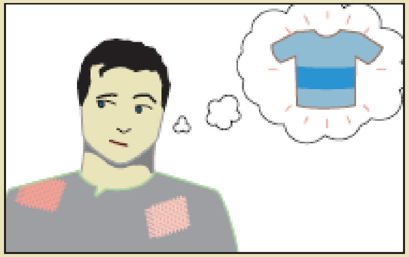
External Event Checklist

- External agent or actor wants something resulting in a transaction
  - E.g., Customer buys a product
- External agent or actor wants some information
  - E.g., Customer wants to know product details
- External data changed and needs to be updated
  - E.g., Customer has new address and phone
- Management wants some information
  - E.g., Sales manager wants update on production plans

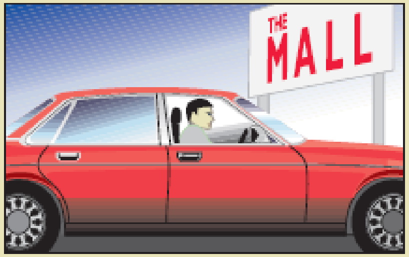
Temporal Event Checklist

- Internal outputs needed at points in time
  - Management reports (summary or exception)
  - Operational reports (detailed transactions)
  - Internal statements and documents (including payroll)
- External outputs needed at points of time
  - Statements, status reports, bills, reminders

Finding the actual event that *directly* affects the system

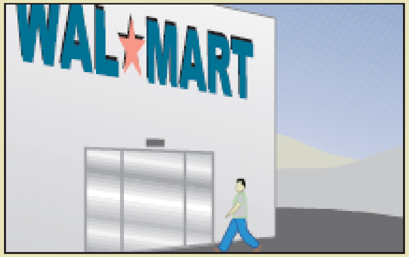


Customer thinks about getting a new shirt

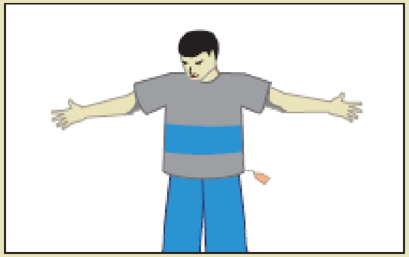


Customer drives to the mall

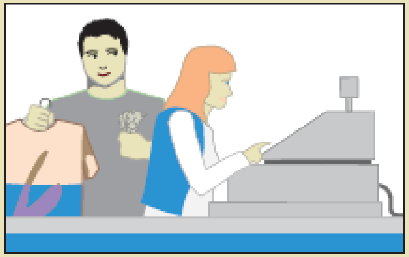
Customer tries on a shirt at Sears



Customer goes to Walmart



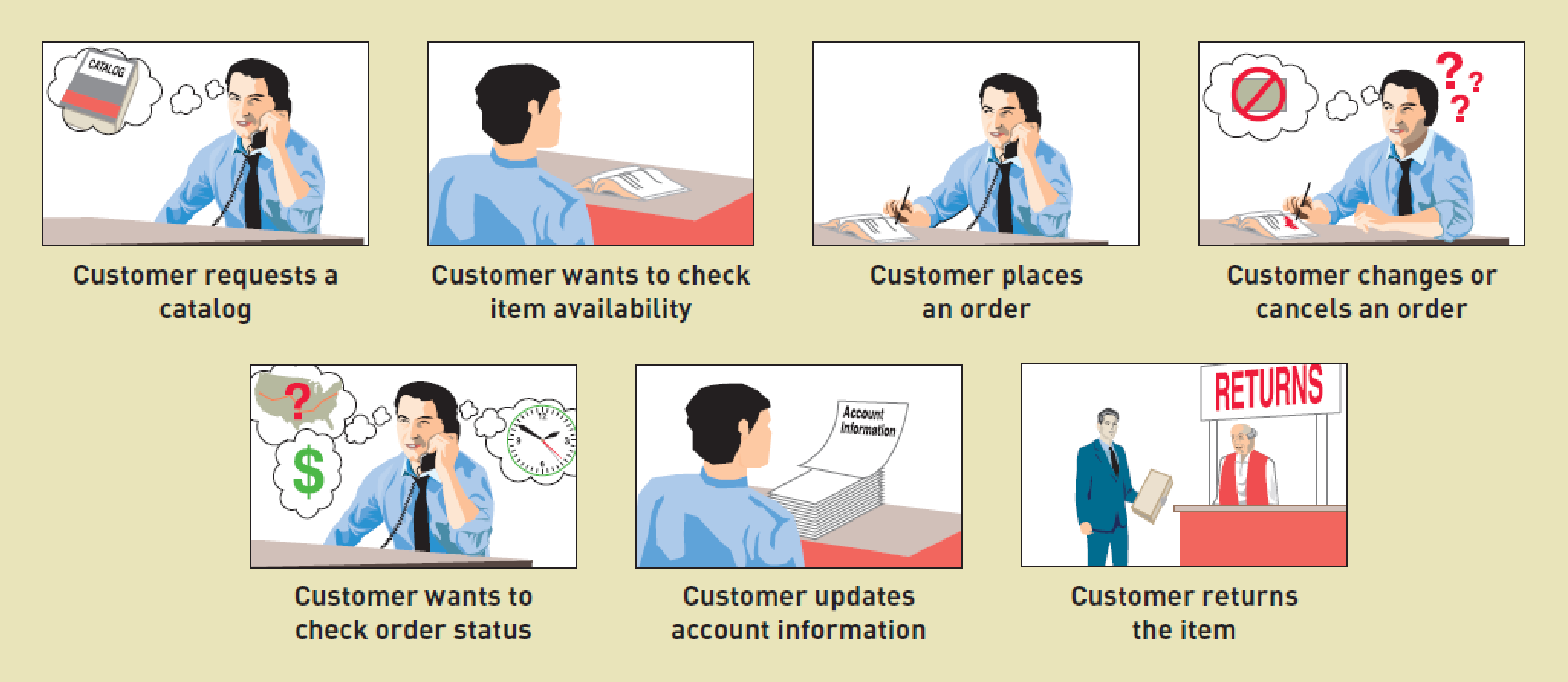
Customer tries on a shirt at Walmart



Customer buys a shirt

(the event that directly affects the system!)

Tracing a sequence of transactions resulting in many events



Customer requests a catalog

Customer wants to check item availability

Customer places an order

Customer changes or cancels an order

Customer wants to check order status

Customer updates account information

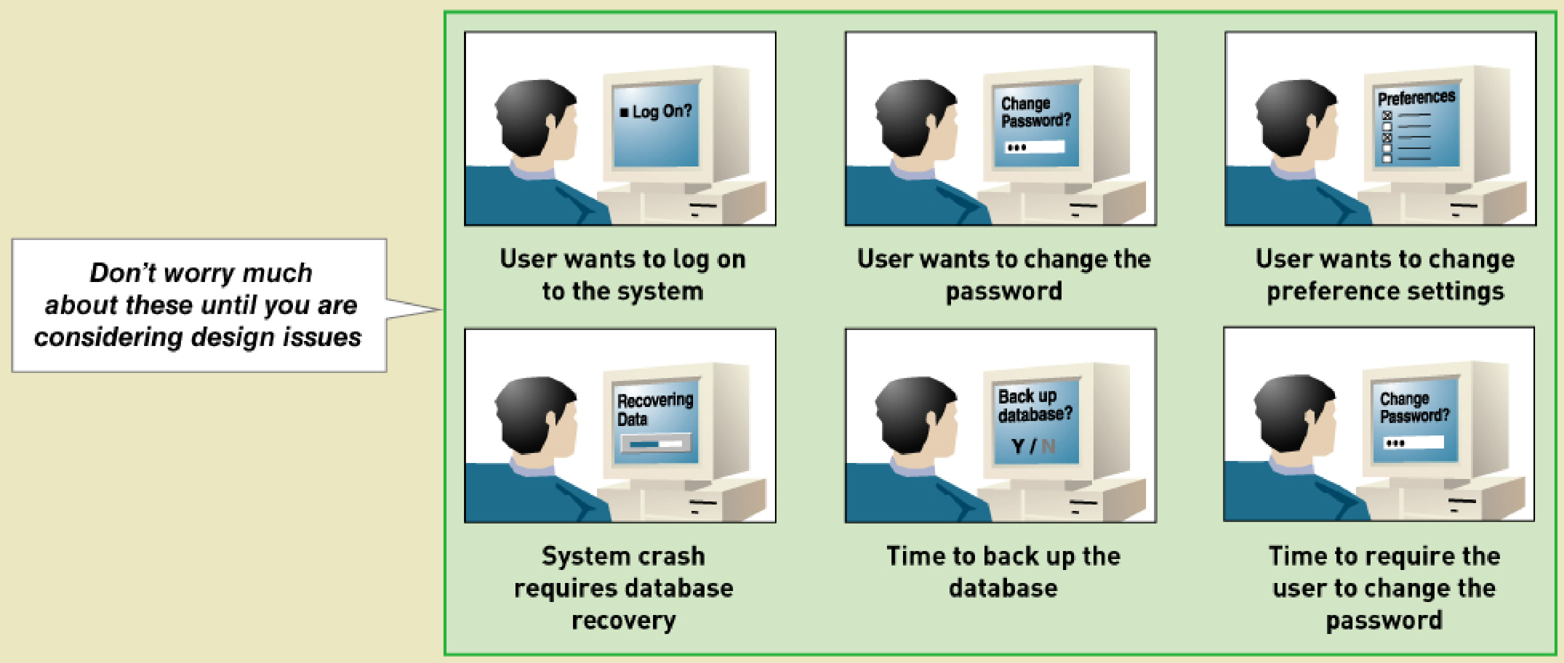
Customer returns the item

Technology-Dependent Events and System Control

- **System Control** – checks or safety procedures put in place to protect the integrity of the system (e.g., logon, database backup)
- **Perfect technology assumption** – events should be included during analysis only if the system would be required to respond under perfect conditions (e.g., no breakdown, no operational mistake).

Technology-Dependent Events and System Control

- Don’t worry about functions built into system because of limits in technology and people. Wait until design.



Don’t worry much about these until you are considering design issues

User wants to log on to the system

User wants to change the password

User wants to change preference settings

System crash requires database recovery

Time to back up the database

Time to require the user to change the password

Event Decomposition Technique: Specific Steps

- For each external event, identify and name the use case that the system requires
- For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case

Event Decomposition Technique: Specific Steps

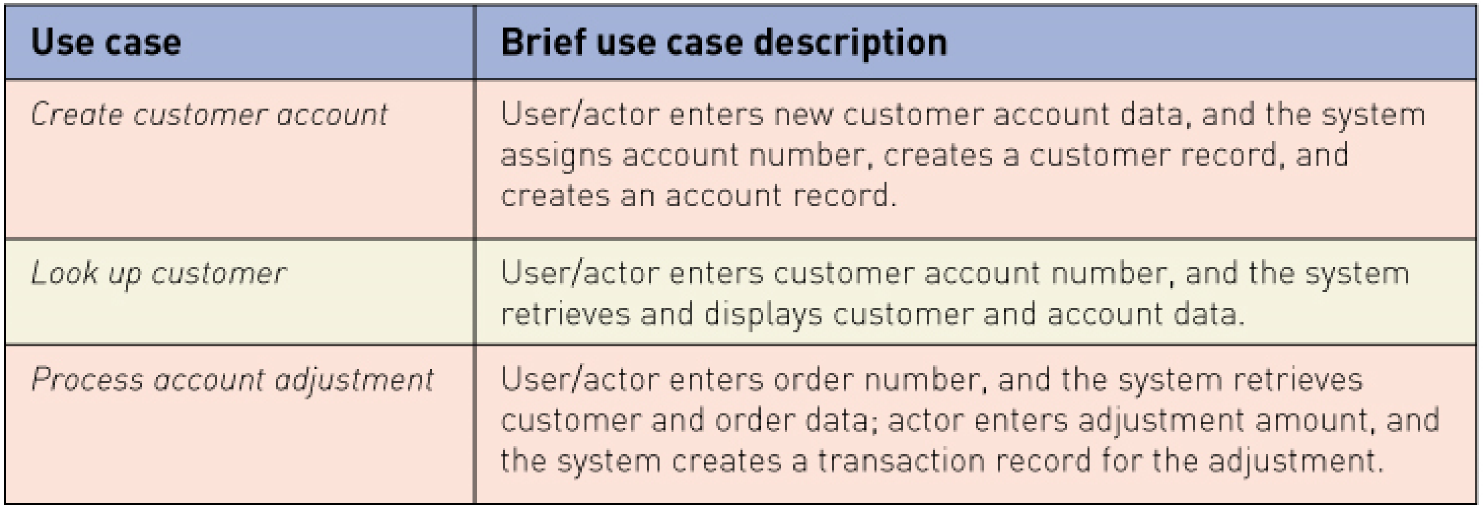
- For each state event (esp. for a real-time system in which devices or internal state changes trigger use cases), identify and name the use case that the system requires and then define the state change
- When events and use cases are defined, check to see if they are required by using the perfect technology assumption
  - *Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in later (in the design stage)*

Event Decomposition Technique: Strength

- Events are broader than user goal: Capture temporal and state events
- Help decompose at the right level of analysis: an Elementary Business Process (EBP)
- Uses perfect technology assumption to make sure functions that support the users work are identified and not additional functions for security and system controls

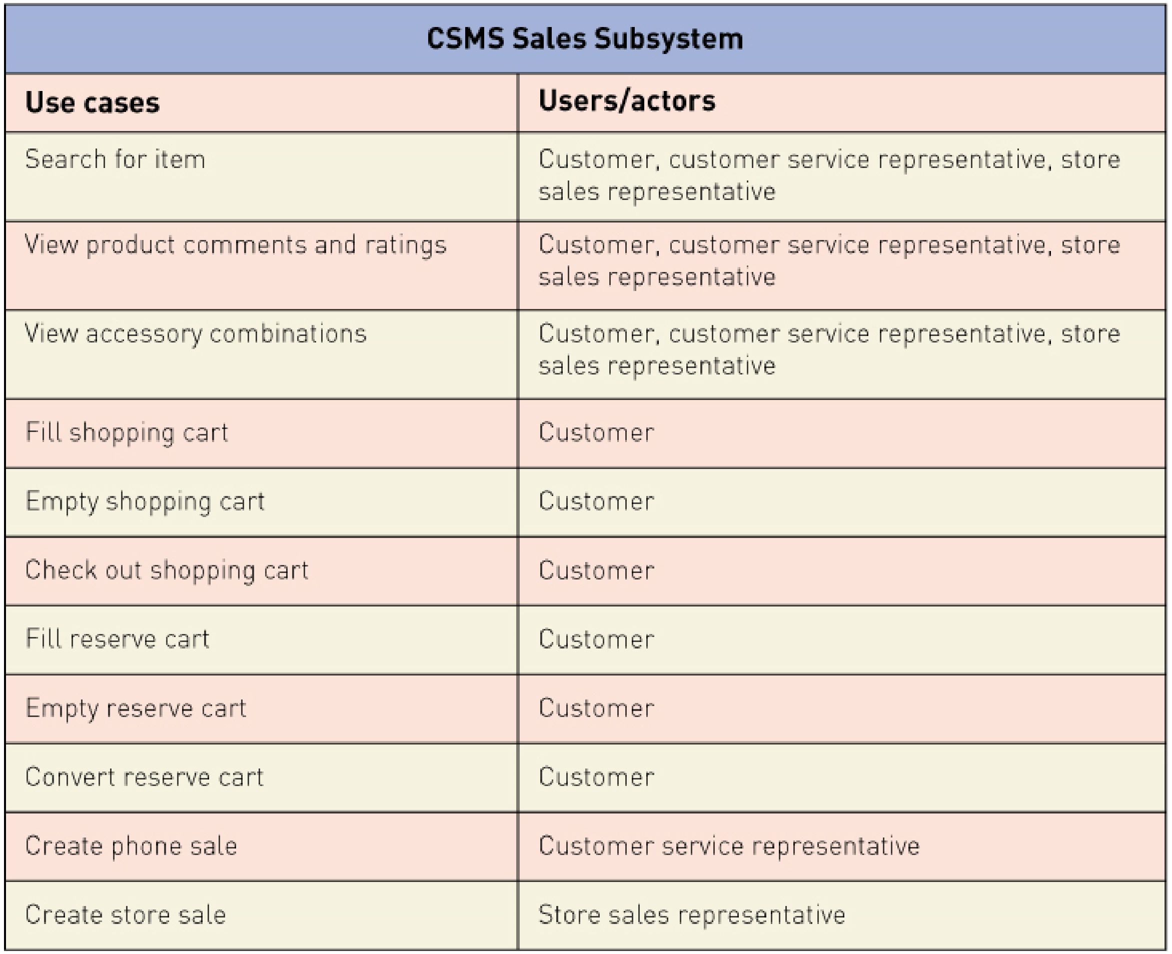
Use Cases and Brief Use Case Descriptions

- Brief use case description is often a one sentence description showing the main steps in a use case



Use case	Brief use case description
Create customer account	User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record.
Look up customer	User/actor enters customer account number, and the system retrieves and displays customer and account data.
Process account adjustment	User/actor enters order number, and the system retrieves customer and order data; actor enters adjustment amount, and the system creates a transaction record for the adjustment.

Use Cases and Users/actors

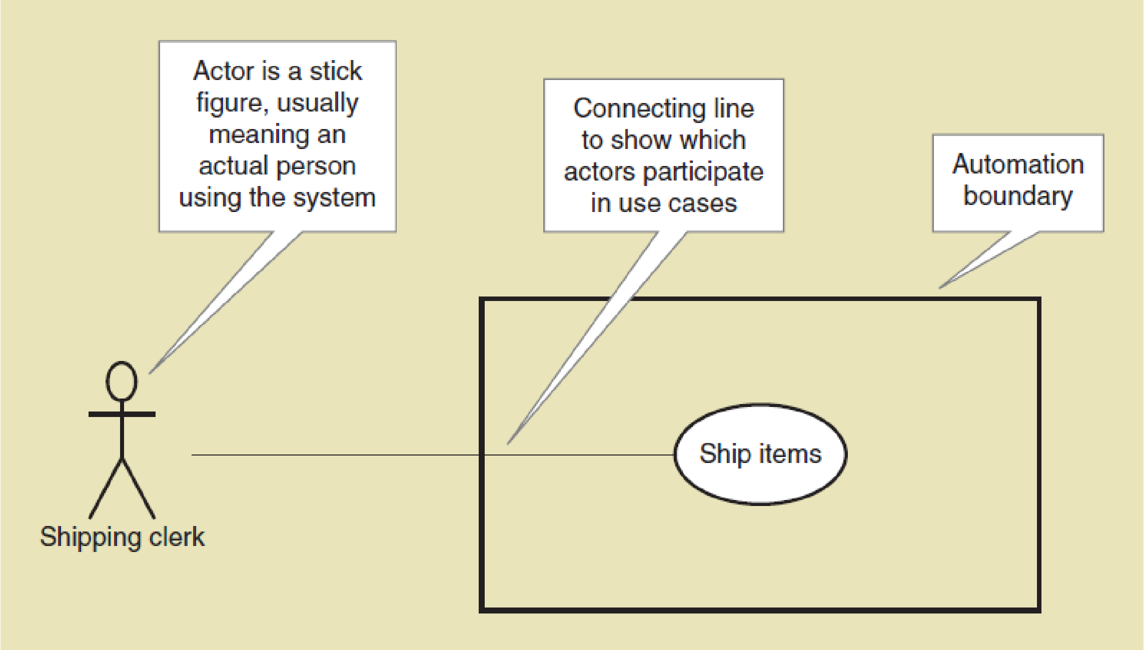


<table>
<tr><th colspan="2">CSMS Sales Subsystem</th></tr>
<tr><th>Use cases</th><th>Users/actors</th></tr>
<tr><td>Search for item</td><td>Customer, customer service representative, store sales representative</td></tr>
<tr><td>View product comments and ratings</td><td>Customer, customer service representative, store sales representative</td></tr>
<tr><td>View accessory combinations</td><td>Customer, customer service representative, store sales representative</td></tr>
<tr><td>Fill shopping cart</td><td>Customer</td></tr>
<tr><td>Empty shopping cart</td><td>Customer</td></tr>
<tr><td>Check out shopping cart</td><td>Customer</td></tr>
<tr><td>Fill reserve cart</td><td>Customer</td></tr>
<tr><td>Empty reserve cart</td><td>Customer</td></tr>
<tr><td>Convert reserve cart</td><td>Customer</td></tr>
<tr><td>Create phone sale</td><td>Customer service representative</td></tr>
<tr><td>Create store sale</td><td>Store sales representative</td></tr>
</table>

Use Case Diagrams

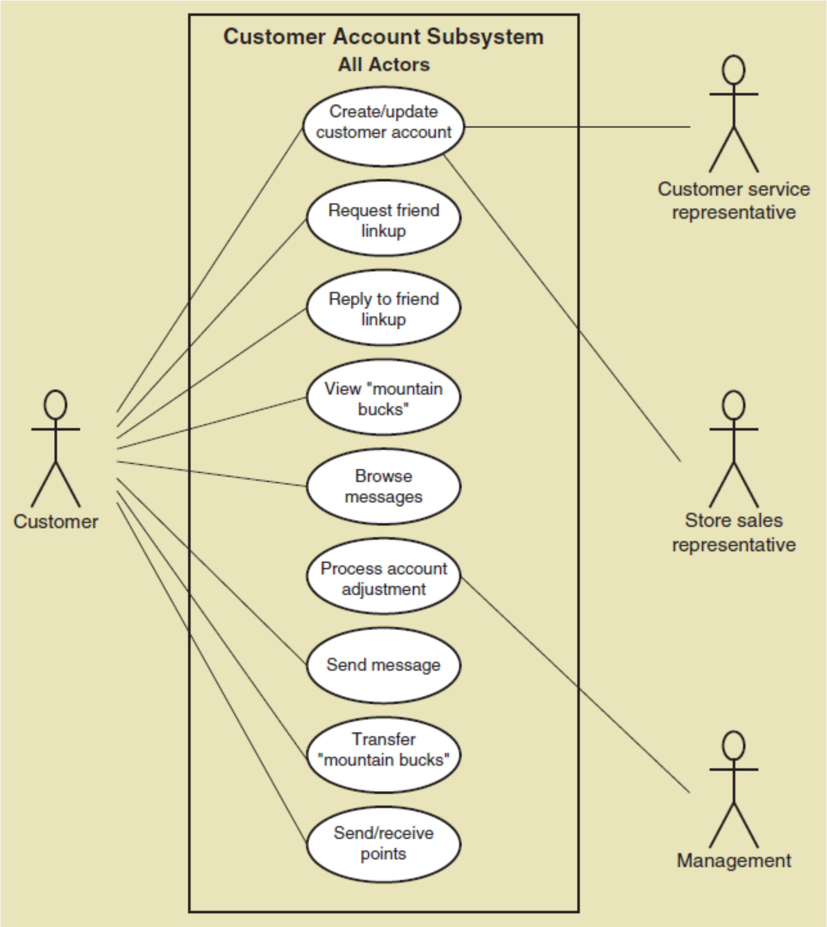
- **Use case diagram** — a UML model used to graphically show uses cases and their relationships to actors
  - *Recall UML is Unified Modelling Language, the standard for diagrams and terminology for developing information systems*
- Actor is the UML name for an end user
- **Automation boundary** — the boundary between the computerised portion of the application and the users who operate the application

Use Case Diagrams Symbols


Actor is a stick figure, usually meaning an actual person using the system
Connecting line to show which actors participate in use cases
Automation boundary
Ship items
Shipping clerk

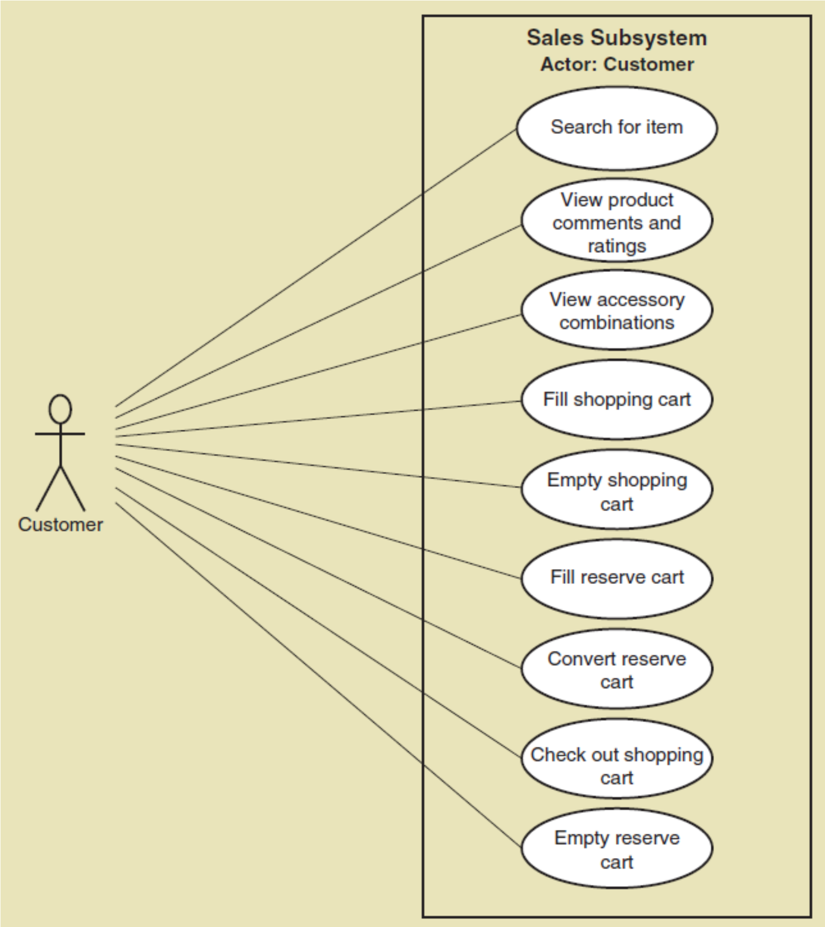
Use Case Diagrams

Draw for each subsystem

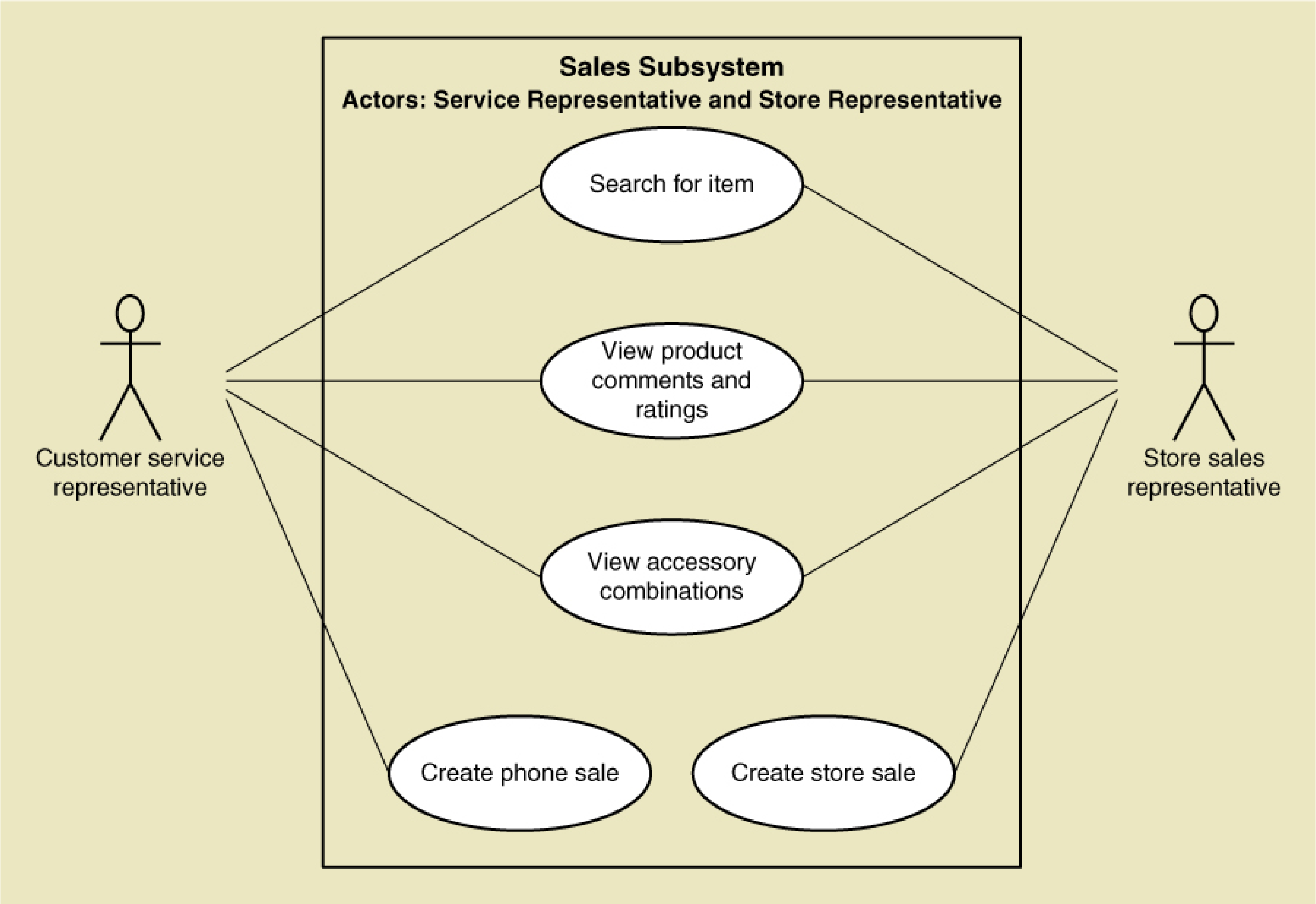

Customer Account Subsystem
All Actors
Create/update customer account
Request friend linkup
Reply to friend linkup
View "mountain bucks"
Browse messages
Process account adjustment
Send message
Transfer "mountain bucks"
Send/receive points
Customer
Customer service representative
Store sales representative
Management

Use Case Diagrams

Draw for a single actor, such as customer

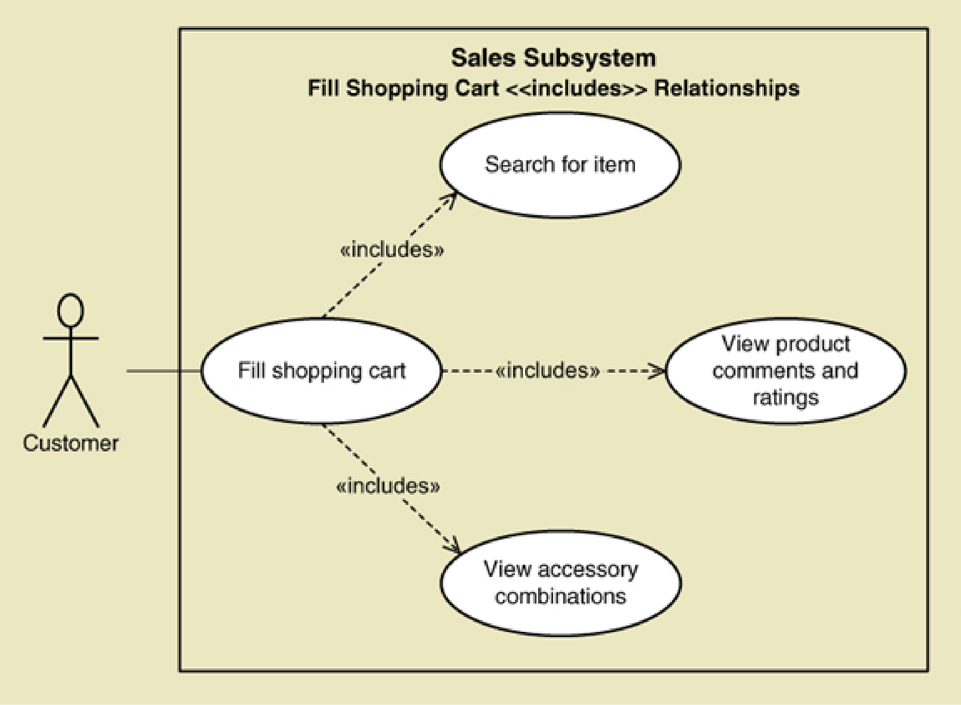


Use Case Diagrams
Draw for internal RMO actors


Sales Subsystem
Actors: Service Representative and Store Representative
Search for item
View product comments and ratings
View accessory combinations
Create phone sale
Create store sale
Customer service representative
Store sales representative

Use Case Diagrams— The <<Includes>> relationship

- A relationship between use cases where one use case is stereotypically included within the other use case— like a called subroutine. Arrow points to subroutine


Sales Subsystem
Fill Shopping Cart <<includes>> Relationships
Search for item
«includes»
Customer
Fill shopping cart
«includes»
View product comments and ratings
«includes»
View accessory combinations

Use Case Diagrams: Steps

1. Identify all the stakeholders and users who would benefit by seeing a use case diagram
2. Determine what each stakeholder or user needs to review in a use case diagram: for each subsystem, for each type of user, for use cases that are of interest
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram (can use many CASE software tools)
4. Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users

Use Case Descriptions

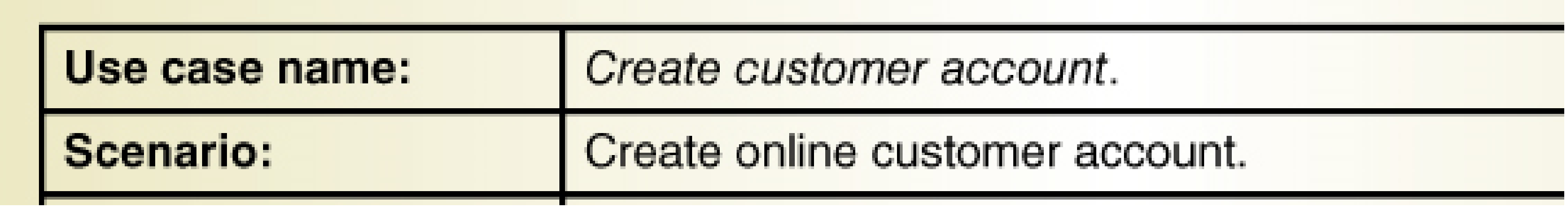
- Write a *fully developed use case description* for more complex use cases
- Typical use case description templates (full) include:
  - Use case name
  - Scenario (if needed)
  - Triggering event
  - Brief description
  - Actors
  - Related use cases (<<includes>>)
  - Stakeholders
  - Preconditions
  - Postconditions
  - Flow of activities
  - Exception conditions

Use Case Descriptions

- Write a *fully developed use case description* for more complex use cases
- Typical use case description templates (full) include:
    - Use case name
    - **Scenario (if needed)**
    - **Triggering event**
    - Brief description
    - Actors
    - **Related use cases (<<includes>>)**
    - **Stakeholders**
    - **Preconditions**
    - **Postconditions**
    - **Flow of activities**
    - **Exception conditions**

Use Case Descriptions

- **Use Name:** a Verb-Noun Phrase
- **Scenarios**: a unique set of internal activities within a use case and represents a unique path through the use case.
  - Several variations of the business steps exist within a single use case. There different flows of activities are called scenarios or <u>use case instances</u>.
  - Example:



Use case name:	*Create customer account.*
Scenario:	Create online customer account.

Use Case Descriptions

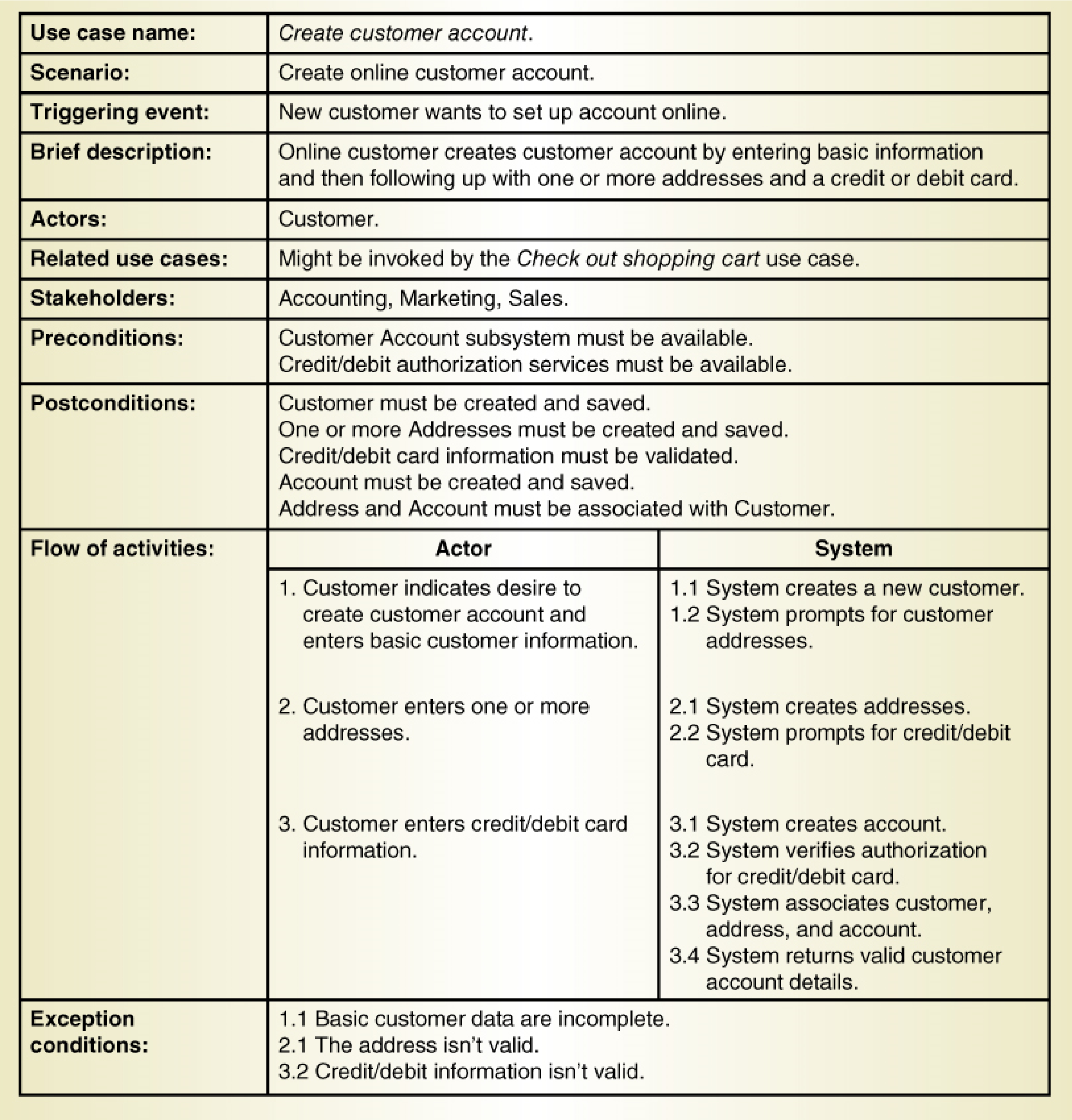
- **Short description:** one sentence brief description of the use case
- **Triggering event:** the event that triggers the use case.
- **Actors:** a person who uses the system (use case) called an actor
- **Related use case:** identifies other use cases and the way they are related to this use case.
  - E.g., **<<includes>>** relationships in use case diagrams
- **Stakeholders:** who are interested parties other than specific actors. They might be users who don't actually invoke the use case but who have an interest in results produced from the use case.

Use Case Descriptions

- **Pre-conditions:** a condition that must be true before a use case begins
  - Can be the subsystem requirements
- **Post-conditions:** what must be true upon the successful completion of a use case
  - Can be used for design and testing
- **Flow of activities:** the detailed flow of activities of the use case (usually one for the actor and one or more for the system or subsystems).
- **Exception conditions:** details the alternative activities and exception conditions.
  - Flexibility of usage
  - Incorrect user processing

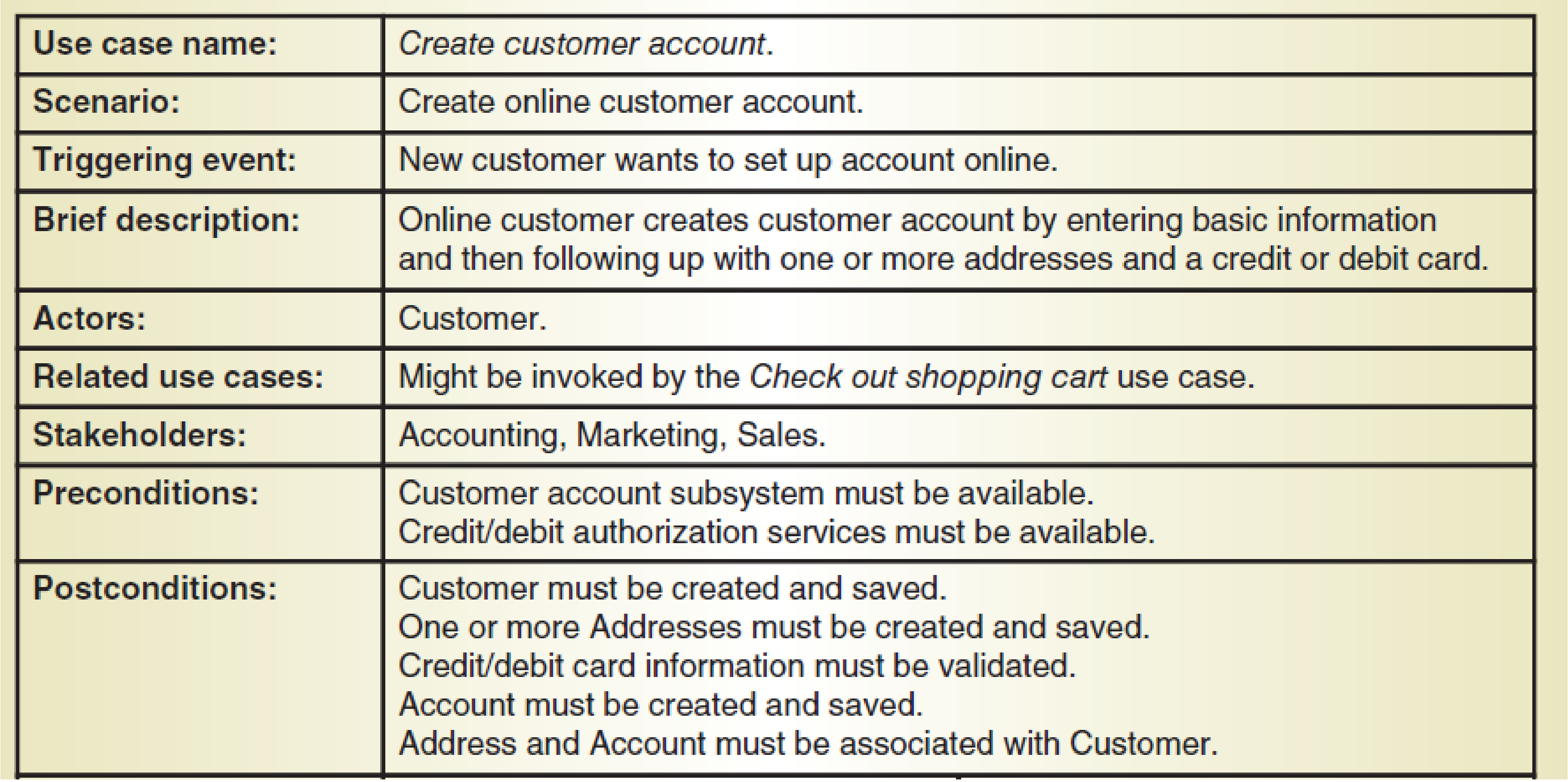
Fully Developed Use Case Description

Use case: *Create customer account*



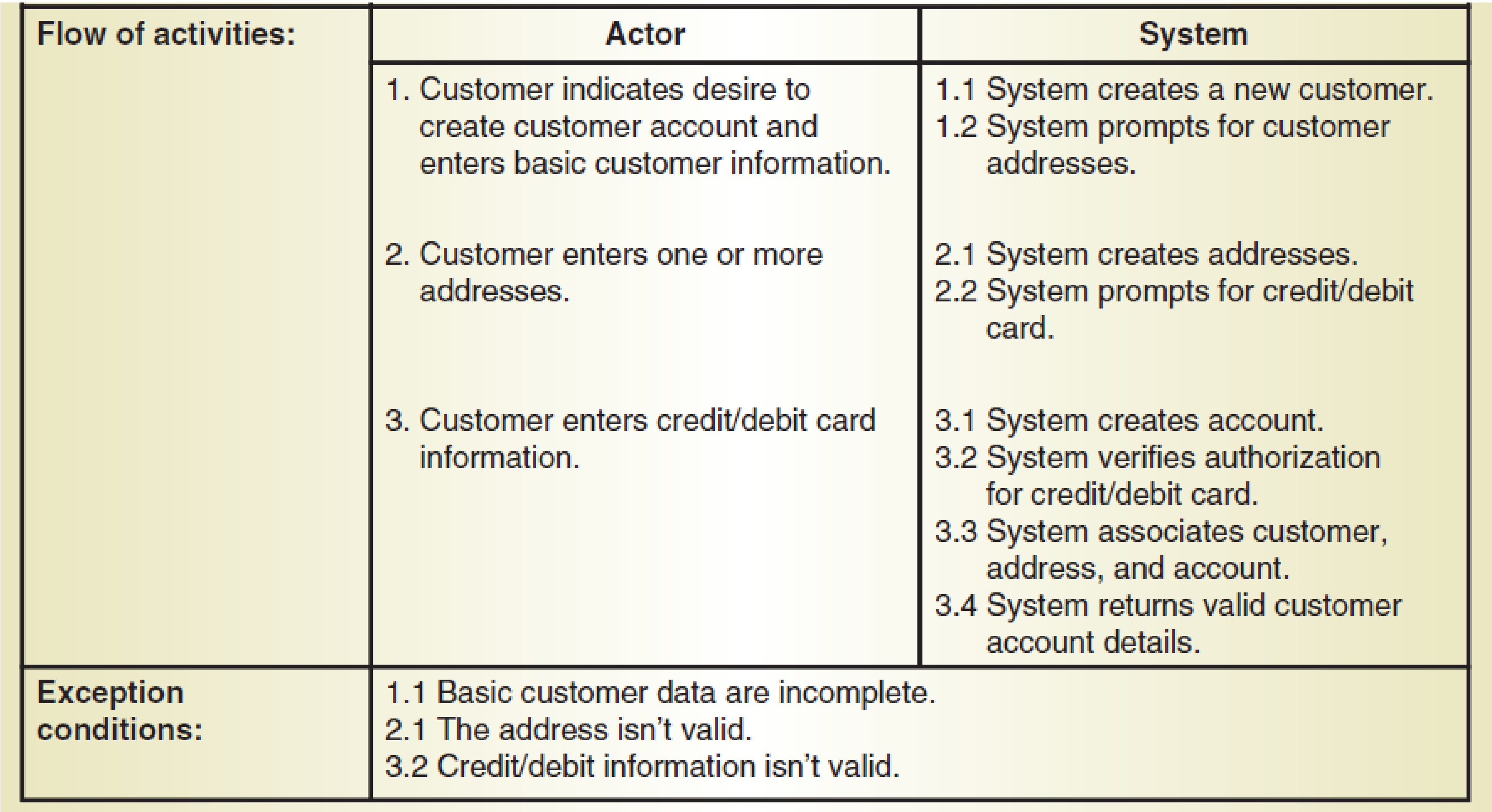
Use case name:	*Create customer account.*	
Scenario:	Create online customer account.	
Triggering event:	New customer wants to set up account online.	
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.	
Actors:	Customer.	
Related use cases:	Might be invoked by the *Check out shopping cart* use case.	
Stakeholders:	Accounting, Marketing, Sales.	
Preconditions:	Customer Account subsystem must be available. Credit/debit authorization services must be available.	
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.	
Flow of activities:	**Actor**	**System**
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

Fully Developed Use Case Description *Create customer account* (part 1)



Use case name:	*Create customer account.*
Scenario:	Create online customer account.
Triggering event:	New customer wants to set up account online.
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.
Actors:	Customer.
Related use cases:	Might be invoked by the *Check out shopping cart* use case.
Stakeholders:	Accounting, Marketing, Sales.
Preconditions:	Customer account subsystem must be available. Credit/debit authorization services must be available.
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.

Fully Developed Use Case Description *Create customer account* (part 2)

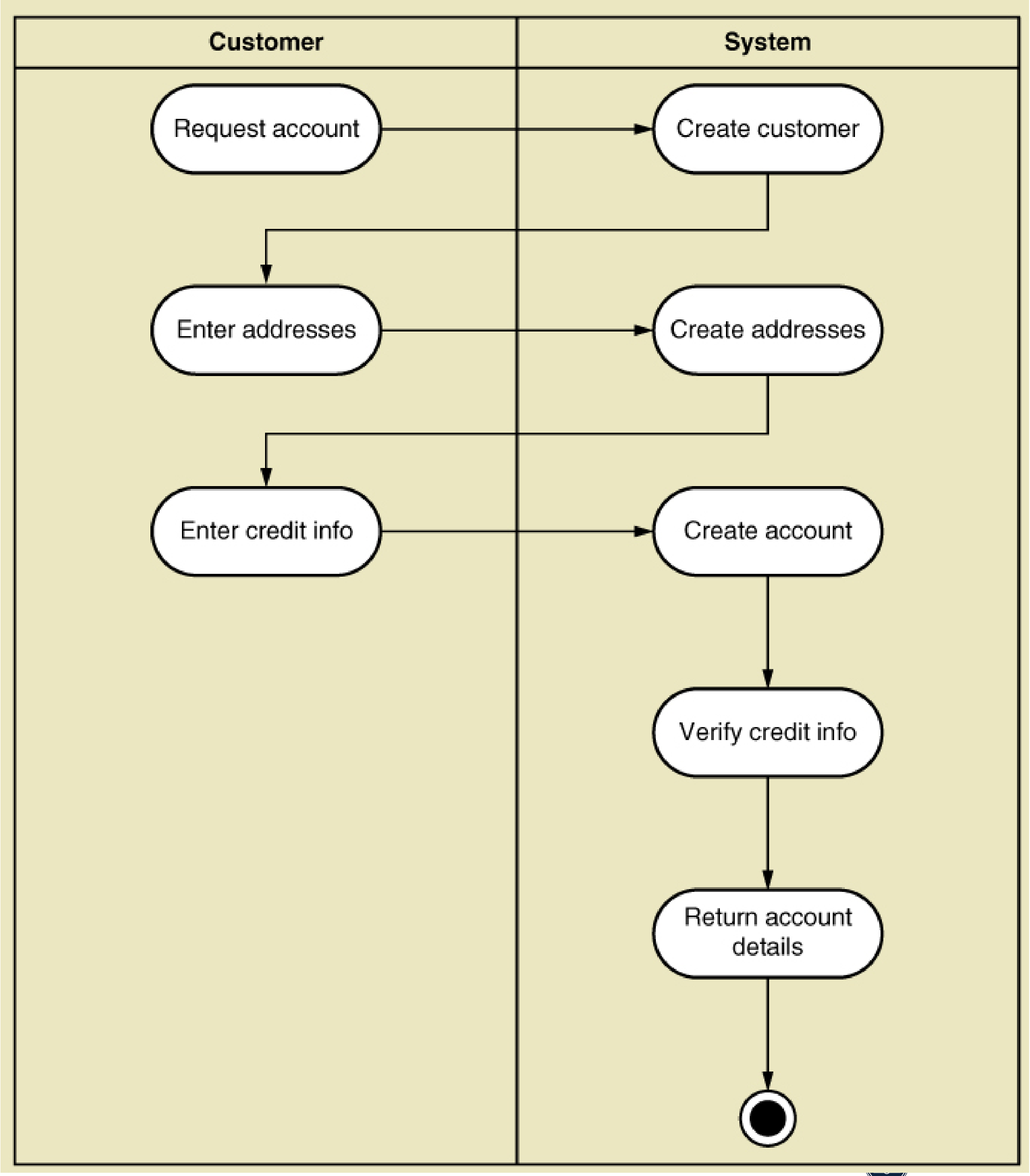


Flow of activities:	Actor	System
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	

UML Activity Diagram for Use Case

Create Customer Account

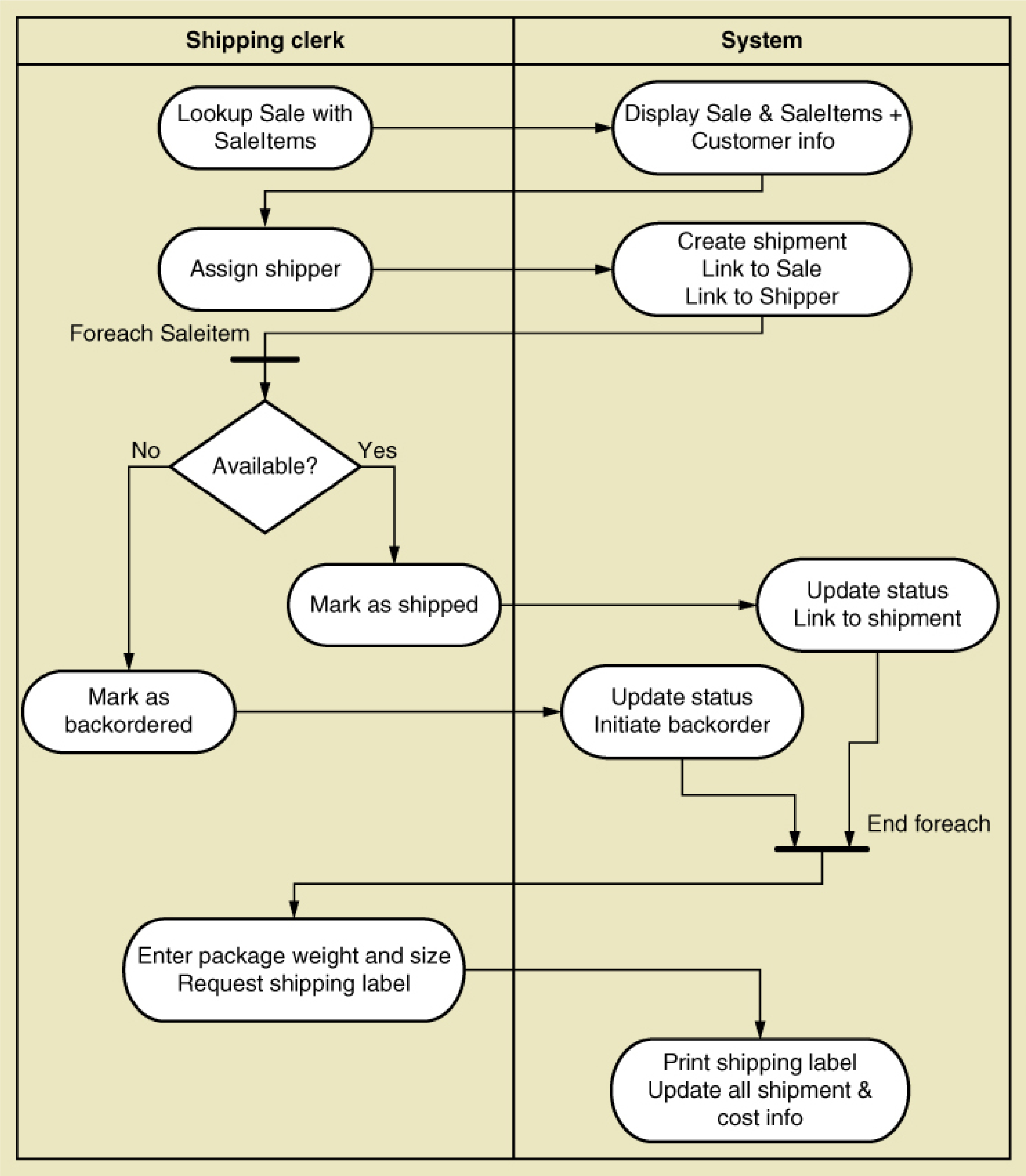
- this shows the flow of activities



UML Activity Diagram for Use Case

Ship Items

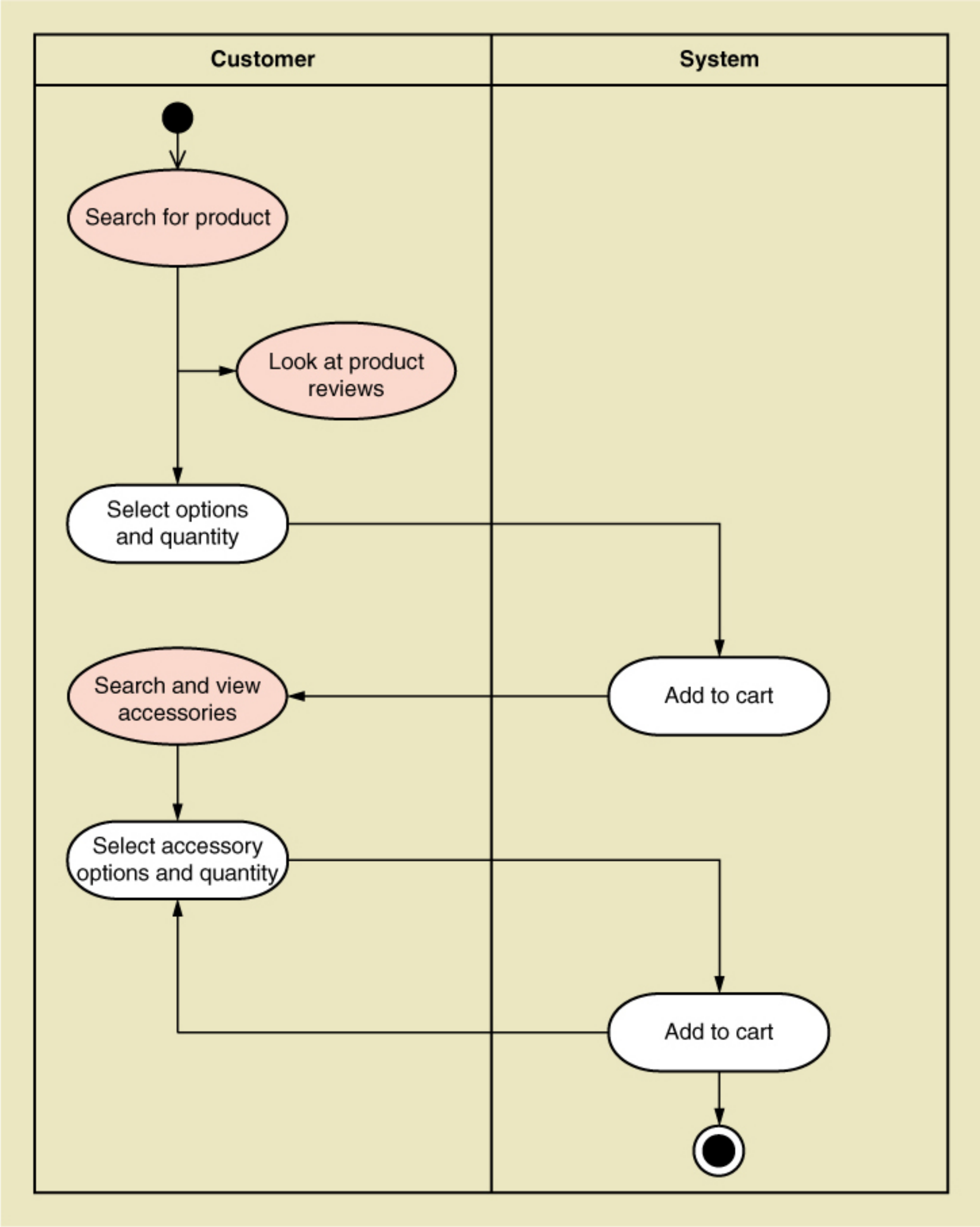
- Note that the synchronization bar indicates a loop


Shipping clerk
System
Lookup Sale with SaleItems
Display Sale & SaleItems + Customer info
Assign shipper
Create shipment Link to Sale Link to Shipper
Foreach Saleitem
No
Available?
Yes
Mark as shipped
Update status Link to shipment
Mark as backordered
Update status Initiate backorder
End foreach
Enter package weight and size Request shipping label
Print shipping label Update all shipment & cost info

UML Activity Diagram for Use Case

Fill Shopping Cart

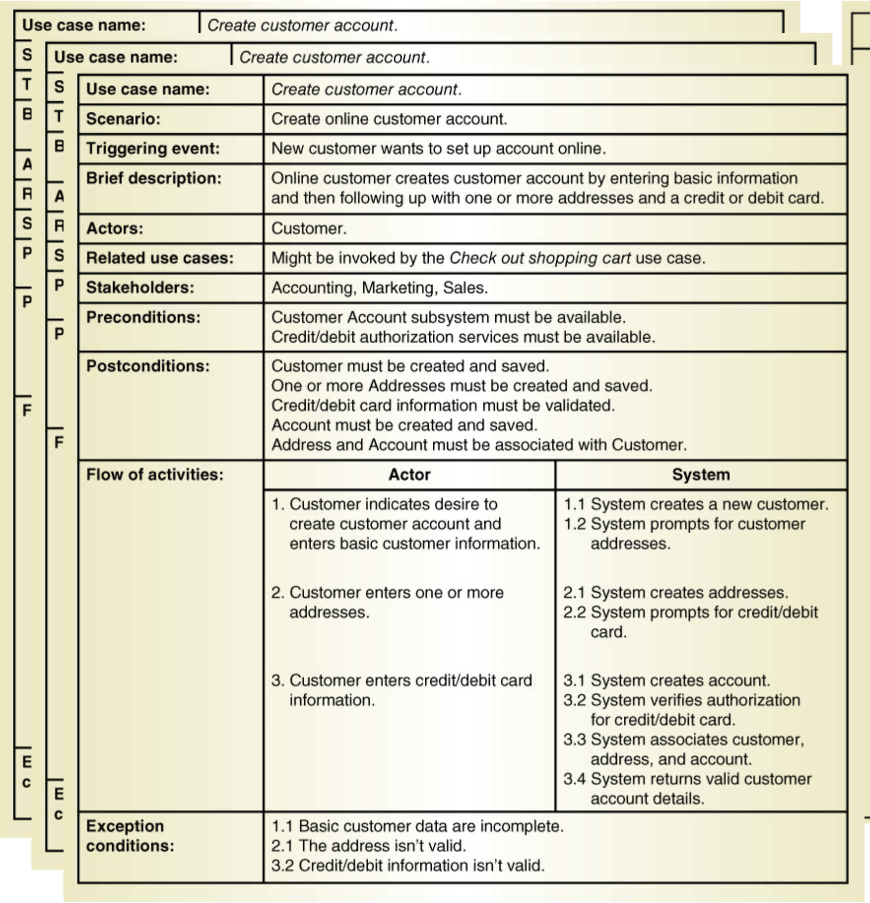
- Note that this activity diagram has references to “external” use cases
- <<includes>> relationships


Customer
System
Search for product
Look at product reviews
Select options and quantity
Add to cart
Search and view accessories
Select accessory options and quantity
Add to cart

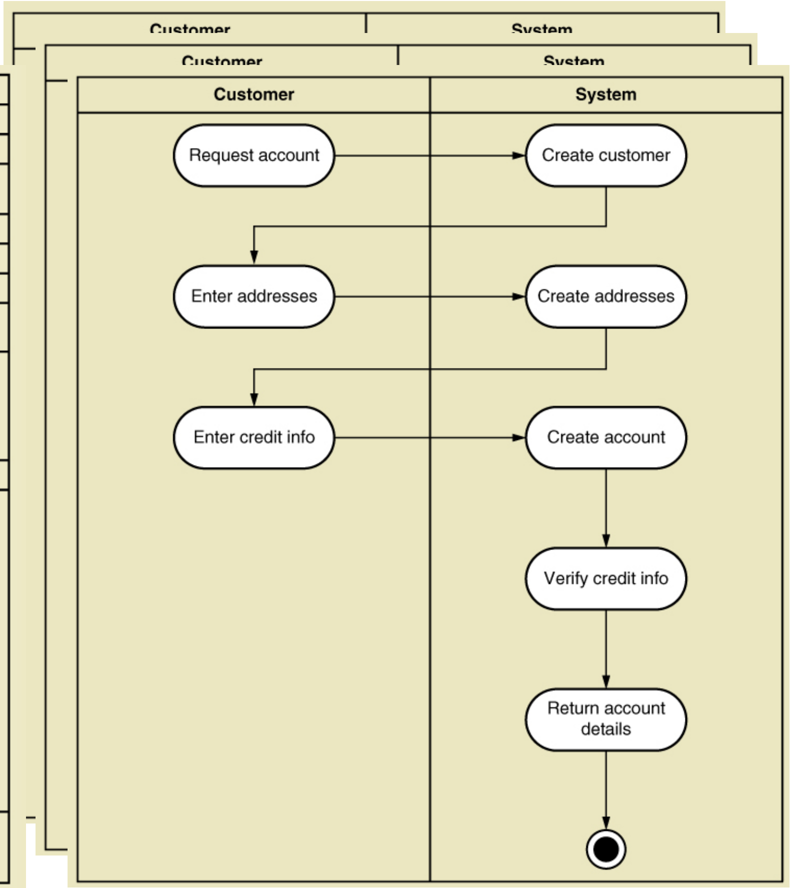
UNIVERSITY
OF WOLLONGONG
AUSTRALIA

Put Together: Use Case Modelling

Use case	Brief use case description
Create customer account	User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record.
Look up customer	User/actor enters customer account number, and the system retrieves and displays customer and account data.
Process account adjustment	User/actor enters order number, and the system retrieves customer and order data; actor enters adjustment amount, and the system creates a transaction record for the adjustment.



Use case name:	*Create customer account.*	
Scenario:	Create online customer account.	
Triggering event:	New customer wants to set up account online.	
Brief description:	Online customer creates customer account by entering basic information and then following up with one or more addresses and a credit or debit card.	
Actors:	Customer.	
Related use cases:	Might be invoked by the *Check out shopping cart* use case.	
Stakeholders:	Accounting, Marketing, Sales.	
Preconditions:	Customer Account subsystem must be available. Credit/debit authorization services must be available.	
Postconditions:	Customer must be created and saved. One or more Addresses must be created and saved. Credit/debit card information must be validated. Account must be created and saved. Address and Account must be associated with Customer.	
Flow of activities:	**Actor**	**System**
	1. Customer indicates desire to create customer account and enters basic customer information.	1.1 System creates a new customer. 1.2 System prompts for customer addresses.
	2. Customer enters one or more addresses.	2.1 System creates addresses. 2.2 System prompts for credit/debit card.
	3. Customer enters credit/debit card information.	3.1 System creates account. 3.2 System verifies authorization for credit/debit card. 3.3 System associates customer, address, and account. 3.4 System returns valid customer account details.
Exception conditions:	1.1 Basic customer data are incomplete. 2.1 The address isn't valid. 3.2 Credit/debit information isn't valid.	



Summary

- A user story is one short sentence in the every-day language of the end user that states what a user does as part of the daily work
- A use case is the activity the system performs in response to a request by a user
- A persona is a typical user that performs a “canonical” use case with the system.
- Two techniques for identifying use cases are the User Goal Technique and the Event Decomposition Technique
- Events include external, temporal and state events.
- An elementary business process (EBP) is the most fundamental task in a business process, usually performed by one actor in response to one event.

Summary

- A use case description is a textual model that lists and describes the processing details for a use case. It can be a brief or full use case description.
- A scenario is a unique set of internal activities within a use case.
- Preconditions and postconditions must hold before and after a use case occurs, respectively.
- An activity diagram is a UML model that demonstrates the flow of activities happening between the actor and the system.

U O W

Questions

UNIVERSITY
OF WOLLONGONG
AUSTRALIA